

ESP8266 HSPI Host Multi-device API



Version 1.1
Copyright © 2016

About This Guide

This document introduces how to operate multi user devices via HSPI host, how to connect, as well as details on API functions, in order to simplify operations between different user devices.

Note:

Please refer to documents “8N-ESP8266_SPI_Communication_User_Guide_CN_v0.1” and “8O-ESP8266_SPI_Overlap_&_Display_Application_Guide_CN_v0.1” for information on configuration.

This document is structured as follows:

Chapter	Title	Subject
Chapter 1	Function Overview	Provides the overview of functions of SPI and HSPI.
Chapter 2	Hardware Connection	Illustrates different ways of hardware connection.
Chapter 3	API Description	Describes APIs.

Release Notes

Date	Version	Release notes
2015.05	V1.0	First release.
2016.01	V1.1	Update Chapter 2.

Table of Contents

- 1. Function Overview4
- 2. Hardware Connection5
- 3. API Description.....6

1. Function Overview

ESP8266 encapsulates two SPI (Serial Peripheral Interfaces) bus segments, shortly named SPI and HSPI. SPI bus is especially used to read CPU programming code from the external Flash, while HSPI bus is used for SPI device communication.

When ESP8266 is working as a host, HSPI bus can operate with three user devices, besides, it also supports one external Flash writing operation. User devices are supported through selection with CS lines. To be more specific,

Mode	Device Name
HSPI Default IO	User device 1
SPI OVERLAP and CS1	User device 2
SPI OVERLAP and CS2	User device 3
SPI OVERLAP and CS0	Flash

In above-mentioned ways of connection, SPI bus shares the same external Flash with HSPI bus. Apart from the memory occupied by programs and related configurations, the rest Flash memory can all be used for reading and writing of user programs.

⚠ Notice:

- Operation with devices via HSPI host implemented by software programming is not supported in the API functions.
- When downloading user programs, the clock frequency of SPI bus used for reading Flash data should be set at 80 MHz. SPI clock frequency should be specified as 80 MHz at SPI OVERLAP and CS1 mode or SPI OVERLAP and CS2 mode.



2. Hardware Connection

Generally speaking, SPI slave devices specify four logic signals: SCLK, MOSI, MISO, and CS.

HSPI bus can operate with three different user devices, the ways of connection are explained below:

Mode	Pin Name of Host ESP8266	SPI bus Signal Line
HSPI Default IO	MTDO	CS
	MTCK	MOSI
	MTDI	MISO
	MTMS	CLK
SPI OVERLAP and CS1	UOTXD	CS1
	SD_CLK	SCLK
	SD_DATA0	MISO
	SD_DATA1	MOSI
SPI OVERLAP and CS2	GPIO0	CS2
	SD_CLK	SCLK
	SD_DATA0	MISO
	SD_DATA1	MOSI

Note:

The pins used when HSPI operates with the Flash in OVERLAP mode is completely the same with that of SPI communication.



3. API Description

Names of the connection modes supported by the system are defined by macro definitions in `\app\include\driver\spi_overlap.h`.

- `HSPI_CS_DEV` (HSPI default IO)
- `SPI_CS1_DEV` (SPI OVERLAP and CS1)
- `SPI_CS2_DEV` (SPI OVERLAP and CS2)

Operation with the Flash is defined as `SPI_CS0_FLASH`. If HSPI operates with two user devices, the API function is shown as below:

```
void hspi_master_dev_init(uint8 dev_no, uint8 clk_polar, uint8 clk_div)
```

Function	This function is used to initialize connection of HSPI host. Altogether four user devices can be operated. If multi devices communicate with the host using SPI communication mode, the function should be called each time when that certain device is operated.
Location	Defined in directory <code>\app\include\driver\spi_overlap.h</code> , implemented in directory <code>\app\driver\spi_overlap.c</code> .
Parameters	<ul style="list-style-type: none">• <code>uint8 dev_no</code>: only <code>HSPI_CS_DEV</code>, <code>SPI_CS1_DEV</code>, <code>SPI_CS2_DEV</code>, and <code>SPI_CS0_FLASH</code> are supported, the corresponding values of these four parameters are 0, 1, 2, and 3 respectively. If the parameter should be other values, ERROR will be printed and the function will be returned.• <code>uint8 clk_polar</code>: clock polarity.<ul style="list-style-type: none">- If the clock polarity is 0, data are captured on the clock's rising edge, and are propagated on a falling edge.- If the clock polarity is 1, data are captured on the clock's falling edge, and are propagated on a rising edge.- If the clock polarity should be other values, ERROR will be printed and the function will be returned.• <code>uint8 clk_div</code>: clock frequency division. 40 MHz is reference frequency, the number of division is <code>clk_div+1</code>. To be more specific, 0 stands for reference frequency, 1 stands for 20 MHz, while 2 stands for 40/3 MHz, and so forth.

⚠ Notice:

ONLY when the clock frequency of SPI bus used for reading Flash data is set at 80 MHz. If the device is defined by `SPI_CS1_DEV` and `SPI_CS2_DEV` via SPI OVERLAP, the clock frequency of host SPI is unadjustable, and should be 80 MHz.



void hspi_dev_sel(uint8 dev_no)

Function	Convert and select host communication devices.
Location	Defined in directory \app\include\driver\spi_overlap.h, implemented in directory \app\driver\spi_overlap.c.
Parameters	uint8 dev_no: only HSPI_CS_DEV, SPI_CS1_DEV, SPI_CS2_DEV, and SPI_CS0_FLASH are supported, the corresponding values of these four parameters are 0, 1, 2, and 3 respectively. If the device has not been initialized, ERROR will be printed and the function will be returned. If it the parameter should be other values, ERROR will be printed and the function will be returned.



Espressif System

IOT Team

<http://bbs.espressif.com>

Disclaimer and Copyright Notice

Information in this document, including URL references, is subject to change without notice.

THIS DOCUMENT IS PROVIDED AS IS WITH NO WARRANTIES WHATSOEVER, INCLUDING ANY WARRANTY OF MERCHANTABILITY, NON-INFRINGEMENT, FITNESS FOR ANY PARTICULAR PURPOSE, OR ANY WARRANTY OTHERWISE ARISING OUT OF ANY PROPOSAL, SPECIFICATION OR SAMPLE. All liability, including liability for infringement of any proprietary rights, relating to use of information in this document is disclaimed. No licenses express or implied, by estoppel or otherwise, to any intellectual property rights are granted herein.

The Wi-Fi Alliance Member logo is a trademark of the Wi-Fi Alliance. The Bluetooth logo is a registered trademark of Bluetooth SIG.

All trade names, trademarks and registered trademarks mentioned in this document are property of their respective owners, and are hereby acknowledged.

Copyright © 2016 Espressif Systems Inc. All rights reserved.